

Docket No. AUS920030826US1

CLAIMS:

What is claimed is:

1. A method, in a data processing system, for automatically identifying performance regression between builds of a computer program based on trace data obtained from a plurality of executions of a first and second build of a computer program, comprising:

obtaining a plurality of call tree data structures corresponding to the trace data for the plurality of executions of the first and second builds of the computer program;

generating a minimized call tree data structure from the plurality of call tree data structures for each of the first and second builds of the computer program, wherein the minimized call tree data structure includes a minimum set of nodes that are consistent between the plurality of call tree data structures;

subtracting the minimized call tree data structure for the second build of the computer program from the minimized call tree data structure of the second computer program to thereby generate a subtracted minimized call tree data structure; and

outputting the subtracted minimized call tree data structure.

2. The method of claim 1, further comprising:

inputting the trace data to an arcflow tool, wherein the arcflow tool generates the plurality of call tree

Docket No. AUS920030826US1

data structures for each of the first and second builds of the computer program based on the trace data.

3. The method of claim 1, wherein the plurality of call tree data structures for each of the first and second builds of the computer program are xtree data structures.

4. The method of claim 1, wherein generating the minimized call tree data structure includes:

 copying a first call tree data structure for a selected build of the computer program; and

 walking a second call tree data structure for the selected build of the computer program over the first call tree data structure to generate the minimized call tree data structure.

5. The method of claim 4, wherein walking the second call tree data structure over the first call tree data structure includes:

 for each node that exists in both the first call tree data structure and the second call tree data structure, generating a node in the minimized call tree data structure and associating values with the node.

6. The method of claim 5, wherein the values associated with the node are values that correspond to the minimum of the values associated with corresponding nodes in the first call tree data structure and the second call tree data structure.

Docket No. AUS920030826US1

7. The method of claim 4, wherein walking the second call tree data structure over the first call tree data structure includes:

for each node that exists in only one of the first call tree data structure and the second call tree data structure, inhibiting creating a node in the minimum call tree data structures.

8. The method of claim 1, wherein subtracting the minimized call tree data structure for the second build from the minimized call tree data structure for the first build to generate a subtracted minimized call tree data structure includes:

copying the minimized call tree data structure for the first build; and

walking the minimized call tree data structure for the second build over the minimized call tree data structure for the first build to generate the subtracted minimized call tree data structure.

9. The method of claim 8, wherein walking the minimized call tree data structure for the second build over the minimized call tree data structure for the first build includes:

for each node that exists in both the minimized call tree data structure for the first build and the minimized call tree data structure for the second build, generating a node in the subtracted minimized call tree data structure by subtracting a minimum base value of the node in the minimized call tree data structure for the second

Docket No. AUS920030826US1

build from a minimum base value of a corresponding node in the minimized call tree data structure for the first build.

10. The method of claim 8, wherein walking the minimized call tree data structure for the second build over the minimized call tree data structure for the first build includes:

for each node that exists in only one of the minimized call tree data structure for the first build and the minimized call tree data structure for the second build, creating a node in the subtracted minimized call tree data structure having a negative minimum base value corresponding to a minimum base value of the node that exists in either of the minimized call tree data structure for the first build or the minimized call tree data structure for the second build.

11. A computer program product in a computer readable medium for automatically identifying performance regression between builds of a computer program based on trace data obtained from a plurality of executions of a first and second build of a computer program, comprising:

first instructions for obtaining a plurality of call tree data structures corresponding to the trace data for the plurality of executions of the first and second builds of the computer program;

second instructions for generating a minimized call tree data structure from the plurality of call tree data structures for each of the first and second builds of the

Docket No. AUS920030826US1

computer program, wherein the minimized call tree data structure includes a minimum set of nodes that are consistent between the plurality of call tree data structures;

third instructions for subtracting the minimized call tree data structure for the second build of the computer program from the minimized call tree data structure of the second computer program to thereby generate a subtracted minimized call tree data structure; and

fourth instructions for outputting the subtracted minimized call tree data structure.

12. The computer program product of claim 11, further comprising:

fifth instructions for inputting the trace data to an arcflow tool, wherein the arcflow tool generates the plurality of call tree data structures for each of the first and second builds of the computer program based on the trace data;

13. The computer program product of claim 11, wherein the plurality of call tree data structures for each of the first and second builds of the computer program are xtree data structures.

14. The computer program product of claim 11, wherein the second instructions for generating the minimized call tree data structure include:

Docket No. AUS920030826US1

instructions for copying a first call tree data structure for a selected build of the computer program; and

instructions for walking a second call tree data structure for the selected build of the computer program over the first call tree data structure to generate the minimized call tree data structure.

15. The computer program product of claim 14, wherein the instructions for walking the second call tree data structure over the first call tree data structure include:

for each node that exists in both the first call tree data structure and the second call tree data structure, instructions for generating a node in the minimized call tree data structure and associating values with the node.

16. The computer program product of claim 15, wherein the values associated with the node are values that correspond to the minimum of the values associated with corresponding nodes in the first call tree data structure and the second call tree data structure.

17. The computer program product of claim 14, wherein the instructions for walking the second call tree data structure over the first call tree data structure includes:

for each node that exists in only one of the first call tree data structure and the second call tree data

Docket No. AUS920030826US1

structure, instructions for inhibiting creating a node in the minimum call tree data structures.

18. The computer program product of claim 11, wherein the third instructions for subtracting the minimized call tree data structure for the second build from the minimized call tree data structure for the first build to generate a subtracted minimized call tree data structure include:

instructions for copying the minimized call tree data structure for the first build; and

instructions for walking the minimized call tree data structure for the second build over the minimized call tree data structure for the first build to generate the subtracted minimized call tree data structure.

19. The computer program product of claim 18, wherein the instructions for walking the minimized call tree data structure for the second build over the minimized call tree data structure for the first build include:

for each node that exists in both the minimized call tree data structure for the first build and the minimized call tree data structure for the second build, instructions for generating a node in the subtracted minimized call tree data structure by subtracting a minimum base value of the node in the minimized call tree data structure for the second build from a minimum base value of a corresponding node in the minimized call tree data structure for the first build.

Docket No. AUS920030826US1

20. The computer program product of claim 18, wherein the instructions for walking the minimized call tree data structure for the second build over the minimized call tree data structure for the first build include:

for each node that exists in only one of the minimized call tree data structure for the first build and the minimized call tree data structure for the second build, instructions for creating a node in the subtracted minimized call tree data structure having a negative minimum base value corresponding to a minimum base value of the node that exists in either of the minimized call tree data structure for the first build or the minimized call tree data structure for the second build.

21. An apparatus for automatically identifying performance regression between builds of a computer program based on trace data obtained from a plurality of executions of a first and second build of a computer program, comprising:

means for obtaining a plurality of call tree data structures corresponding to the trace data for the plurality of executions of the first and second builds of the computer program;

means for generating a minimized call tree data structure from the plurality of call tree data structures for each of the first and second builds of the computer program, wherein the minimized call tree data structure includes a minimum set of nodes that are consistent between the plurality of call tree data structures;

Docket No. AUS920030826US1

means for subtracting the minimized call tree data structure for the second build of the computer program from the minimized call tree data structure of the second computer program to thereby generate a subtracted minimized call tree data structure; and

means for outputting the subtracted minimized call tree data structure.